METHOD AND APPARATUS FOR IMPROVING VIDEO QUALITY OF LOW BIT-RATE VIDEO

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present Application for Patent claims priority to Provisional Application No. 60/456,030 entitled "Method and Apparatus for Improving Video Quality of Low Bit-Rate Video" filed March 17, 2003, and assigned to the assignee hereof and hereby expressly incorporated by reference herein.

[0002] This patent application is related to the co-pending US Patent Application Nos. 10/715,572 and 10/715,573, both entitled "System and Method for Multi-Description Encoding," filed on November 17, 2003, and assigned to the assignee of the present invention.

BACKGROUND

I. Field of Invention

[0003] The invention generally relates to data compression and more particularly to block based compression systems.

II. Description of the Related Art

[0004] Transform coding is a common image compression technique that divides an image into sub-images or blocks for processing. Block-based compression introduces artifacts between block boundaries because blocks are independently coded. Therefore, the transform does not take into consideration the correlation between block boundaries. Accordingly, the technique typically results in low bit rate images that are heavily compressed, but contain severe compression artifacts such as blocking, ringing and motion smear.

[0005] As a result, several post-processing algorithms for deblocking have been proposed to reduce and/or eliminate compression artifacts. However, many involve complex computations and may result in an overall blurring effect on the output image. Other blocking filters do not conserve edge information effectively and are typically complex to implement in hardware. Accordingly, there is need for a more simple and/or effective deblocking process.

SUMMARY

[0006] A method and apparatus for processing images compressed using block based compression may comprise determining whether two blocks are neighboring blocks; determining whether the two neighboring blocks are both subdivided; performing deblocking filter on one or more edge pixels of the two neighboring blocks if both of the two neighboring blocks are not subdivided. Determining whether two neighboring blocks are both subdivided may comprise obtaining variance values of each of the two neighboring blocks; comparing the variance values to a first threshold; and determining whether the two neighboring blocks are both subdivided based upon the comparison of the variance values to the first threshold. Alternatively, determining whether two neighboring blocks are both subdivided also may also comprises obtaining a block size assignment value; and using the block size assignment value to determine whether the two neighboring values are subdivided.

[0007] The method and apparatus may further comprise determining whether one of the two neighboring blocks is subdivided, if both of the two neighboring blocks are not subdivided; using a first deblocking filter on one or more edge pixels of the two neighboring blocks if one of the two neighboring blocks is subdivided; and using a second deblocking filter on one or more edge pixels of the two neighboring blocks if neither of the two neighboring blocks are subdivided.

[0008] The method and apparatus may further comprise obtaining one or more difference values of one or more edge pixels of the two neighboring blocks, if neither of the two neighboring blocks are subdivided; comparing the one or more difference values to a second threshold; and selecting the second deblocking filter based on the comparison of the one or more difference values to the second threshold.

[0009] Obtaining one or more difference values may comprise obtaining difference values between three edge pixels of the two neighboring blocks; and selecting the second deblocking filter may comprise using a Gaussian filter if at least two of the difference values are greater than the second threshold.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Various embodiments will be described in detail with reference to the following drawings in which like reference numerals refer to like elements, wherein:

[0011] Figure 1 is one example of an image compressor;

[0012] Figure 2 is one example of an image decompressor;

[0013] Figure 3 shows an example process for determining whether a block is subdivided;

[0014] Figures 4A to 4D show examples of block subdivision;

[0015] Figure 5 show an example of two neighboring blocks in an image;

[0016] Figure 6 shows an example process for determining whether to use deblocking filter;

[0017] Figure 7 shows another example process for determining whether to use deblocking filter;

[0018] Figures 8A to 8D show the orderings of ABSDCT for a 16x16 block;

[0019] Figures 9A and 9B show examples of block size assignment data;

[0020] Figures 10A and 10B show more examples of block size assignment data;

[0021] Figure 11 shows an example process for generating block size assignment data for ABSDCT;

[0022] Figure 12 is a table showing the different variance threshold values;

[0023] Figure 13 shows some variable definitions; and

[0024] Figure 14 shows an example process for determining whether to use deblocking filter in systems using ABSDCT.

DETAILED DESCRIPTION

[0025] In compression systems using block based Discrete Cosine Transform (DCT), a data stream is divided into pixel blocks and discrete cosine transformed. This block based processing introduces blocking artifacts between block boundaries since the transform does not take into account the correlation between block boundaries and since each block is independently coded.

[0026] Typically, in compression systems using DCT, the size of each data block is fixed. However, there are dynamic image compression techniques capable of offering significant compression while preserving the quality of image signals utilizing adaptively sized blocks and sub-blocks of encoded DCT coefficient data. Such techniques will be called variable block size DCT. One example of variable block size DCT is the adaptive block size discrete cosine transform (ABSDCT) disclosed in U.S. Pat. No. 5,021,891, entitled "Adaptive Block Size Image Compression Method And System." DCT techniques are also disclosed in U.S. Pat. No. 5,107,345, entitled "Adaptive Block Size Image Compression Method And System," and the use of the ABSDCT technique in combination with a Discrete Quadtree Transform technique is

discussed in U.S. Pat No. 5,452,104, entitled "Adaptive Block Size Image Compression Method And System." The adaptive block sizes are chosen to exploit redundancy that exists for information within a frame of image data. ABSDCT will be described later in more detail.

[0027] The embodiments described below reduces artifacts by allowing a simple and effective deblocking process that can easily be implemented in compression systems using block based DCT. The embodiments are especially effective in variable block size DCT.

[0028] In the following description, specific details are given to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific detail. For example, circuits may be shown in block diagrams in order not to obscure the embodiments in unnecessary detail. In other instances, well-known circuits, structures and techniques may be shown in detail in order not to obscure the embodiments.

[0029] It is also noted that the embodiments may be described as a process which is depicted as a flowchart, a flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be re-arranged. A process is terminated when its operations are completed. A process may correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. When a process corresponds to a function, its termination corresponds to a return of the function to the calling function or the main function.

[0030] Figure 1 shows an example of an image compressor 100 and Figure 2 shows an example of an image decompressor 200 that is symmetric to the image compressor 100. Image compressor 100 comprises a variable block size DCT (VBSDCT) module 110, a quantization module 120 and a variable length coding (VLC) module 130. Image decompressor 200 comprises a variable length decoding (VLD) module 210, an inverse quantization module 220 and an inverse VBSDCT module 230. Image decompressor 200 further comprises a deblocking filter module 240 to filter block edges when necessary and a processor 250 to control the deblocking filter module 240.

[0031] Typically, data stream input to image compressor 100 is composed of image frames. An image frame can generally be divided into slices, a slice can be divided into data blocks, and a data block can be divided into pixels which are the smallest units of

an image. Each image frame includes an integer number of slices and each image slice represents the image information for a set of n consecutive scan lines, such as 16 consecutive scan lines. In such case, each data block corresponds to a 16x16 pixel block across the image of the frame. Also, a frame may be separated into even and odd slices, thereby forming even half frame and odd half frame. Moreover, an image pixel can be commonly represented in the Red, Green and Blue (RGB) color component system. However, because the human eye is more sensitive to changes in luminance and less sensitive to changes in chrominance, the YCbCr color space is typically used in video compression to represent image pixels. The YCbCr color space is a linear transformation of the RGB components, where Y is the chrominance component, and Cb and Cr are the color components. If a frame is separated into even/odd frames, an image frame would be made up of three even half frames and three odd half frames corresponding to the components Y, Cb and Cr.

[0032] In the description above, a slice can represent a set of consecutive scan lines other than 16 consecutive scan lines. Also, the data block may be an nxm block, where n is not equal to m, if the block can be subdivided Moreover, a different color space with the same or different number of color components may be used to represent an image pixel. However, a block size of 16x16 pixels and the YCbCr color space will be used below for purposes of explanation.

[0033] Referring back to Figure 1, VBSDCT module 110 converts the digital image information from spatial to frequency domain and generates DCT coefficients with corresponding block size assignment (BSA) information. VBSDCT module 110 divides and processes the digital image information in blocks and sub-blocks as necessary. Figure 3 shows one process 300 to determine whether a block is subdivided. In process 300, the variance value of a DCT block is obtained (310). The variance is then compared to a threshold value TBS set for the size of the block (320). A determination is made whether the block is subdivided based upon the comparison of the variance value to the threshold TBS. Namely, if the variance value is greater than TBS, then the block is subdivided (330 and 340). Otherwise, the block is not subdivided (350). Here, an analogous process may be used on a sub-block to determine whether the sub-block is subdivided. In such case, the variance would be compared to a threshold value set for the size of the sub-block.

[0034] The BSA information indicates how a block is subdivided, if a block is divided. For example, the BSA information may indicate that a 16x16 block is subdivided into

four 8x8 blocks as in Figure 4A or perhaps that an 8x8 block is subdivided into four 4x4 blocks as in Figure 4B. In other systems, the BSA information may indicate that a 16x16 block is subdivided into four 8x8 blocks as shown in Figure 4C. As shown, one of the 8x8 block of Figure 4C is subdivided into four 4x4 blocks and further into four 2x2 blocks as also shown in Figures 4C, based on the system configuration and/or needs. Quantization module 120 then quantizes the DCT coefficients and VLC 130 compresses the quantized DCT coefficients using a variable length coding technique.

[0035] At image decompressor 200, VLD module 210 decompresses compressed image information, quantization module 220 inverse quantizes the decompressed image information and inverse VBSDCT module 230 converts the inverse quantized image information from frequency to spatial domain, using the block size assignment information. Processor 250 determines whether two blocks of the image are neighboring blocks as shown in Figure 5. Processor 250 then determines whether deblocking is necessary for the two neighboring blocks based on the amount of block edge activity or busyness. If deblocking is deemed necessary, one or more common edge pixels of the two neighboring blocks are filtered by deblocking filter module 240. The post-processed image information is then output to a display and/or stored for presentation.

[0036] Figure 6 shows one process 600 for processing images compressed using block based compression. In process 600, a determination is made whether two neighboring blocks are both subdivided (610). Here, the BSA information may be used to determine whether the two neighboring blocks are subdivided. If both of the two neighboring blocks are not subdivided, then deblocking filter is used on one or more edge pixels of the two neighboring blocks (620).

[0037] Figure 7 shows another process 700 for processing images compressed using block based compression. In process 700, a determination is made whether two neighboring blocks are both subdivided (710). If determined that the two neighboring blocks are both subdivided, deblocking filter is not used. However, if both of the two neighboring blocks are not subdivided, i.e. at least one of the two neighboring blocks is not subdivided, a further determination is made whether one of the two neighboring blocks is subdivided (720). If one of the two neighboring blocks is subdivided, then a first deblocking filter is used on one or more pixels of the two neighboring blocks (730). Here, the first deblocking filter may be a two point averaging filter used on two edge pixels of the two neighboring blocks. If neither of the two neighboring blocks are

subdivided, then a second deblocking filter is used on one or more pixels of the two neighboring blocks.

[0038] More particularly, difference values between one or more corresponding edge pixels of the two neighboring blocks are obtained (740). A difference value represents the variance across a block boundary and may be obtained and/or derived using various techniques. A simple first order difference between two corresponding edge pixels of two neighboring blocks may be obtained. In other embodiments, a second order difference may be obtained and used. The one or more difference values are compared to a threshold TD (750). Based on the comparison of the one or more difference values to the threshold TD, the second deblocking filter is selected (760).

[0039] The threshold TD generally depends on the luminance and can be set for ahead of time for different systems and/or different types of images. In one embodiment, the average of the mean values of the two neighboring blocks may be used as the threshold TD. Alternatively, the threshold TD may be the difference in the mean values of the two neighboring blocks. The threshold TD may also be optimized to deal with intensity variations in an image using a scale factor α which is proportional to the contrast ratio defined as follows, where μc is the mean values of the current block and μn is the mean of a block containing the edge pixels used in obtaining the difference values.

$$\alpha = (|\mu_c - \mu_n|)/\mu_n$$

[0040] The value of α ranges from 0 to 1.

[0041] Furthermore, in one embodiment, the difference values between three edge pixels of the two neighboring blocks are obtained and compared with threshold TD. If at least two of the difference values are greater than TD, a Gaussian filter is selected. Namely, if three of the three difference values are greater than TD , a six point Gaussian filter is used on six edge pixels of the two neighboring blocks. If two of the three difference values are greater than TD, then a four point Gaussian filter is used on four edge pixels of the two neighboring blocks. If one of the three difference values is greater than TD, an averaging filter is used on two edge pixels of the two neighboring blocks.

[0042] Referring back to Figure 2, processor 250 can therefore determine whether deblocking is necessary. As discussed above, processor 250 can also select, as the system allows, different deblocking filters depending on the characteristics of the

neighboring blocks. Thus declocking filter module 240 may comprises one or more types of filters such as, but not limited to, an averaging filter and/or Gaussian filter.

[0043] Furthermore, as discussed above, VBSDCT module 110 may be implemented by ABSDCT. Compression techniques using ABSDCT will next be described using a block size of 16x16 pixels. Generally, each of the luminance and chrominance components is passed to a block interleaver (not shown). In one embodiment ass, as shown in Figures 8A to 8D, a 16x16 block is presented to the block interleaver, which orders the image samples within the 16x16 blocks to produce blocks and composite sub-blocks of data for DCT analysis. One 16x16 DCT is applied to a first ordering, four 8x8 DCTs are applied to a second ordering, 16 4x4 DCTs are applied to a third ordering, and 64 2x2 DCTs are applied to a fourth ordering. The DCT operation reduces the spatial redundancy inherent in the image source. After the DCT is performed, most of the image signal energy tends to be concentrated in a few DCT coefficients.

[0044] For the 16x16 block and each sub-block, the transformed coefficients are analyzed to determine the number of bits required to encode the block or sub-block. Then, the block or the combination of sub-blocks that requires the least number of bits to encode is chosen to represent the image segment. For example, two 8x8 sub-blocks, six 4x4 sub-blocks, and eight 2x2 sub-blocks may be chosen to represent the image segment. The chosen block or combination of sub-blocks is then properly arranged in order.

[0045] The transformed coefficients are analyzed and the block or the combination of sub-blocks to represent the image segment is selected. Thus, the block size assignment information that represents the block size assignment within an nxn block is generated. For the 16x16 data block, ABSDCT technique generates data known as PQR information that represents the block size assignment within the 16x16 block. The PQR information is a variable bit width data and describes to what extent a 16×16 block is subdivided. The R-bit of the PQR field represents whether the 16x16 block is subdivided into four 8x8 blocks. As shown in Figure 9A, if the R bit is '0', the block remains whole. In this case no further PQR information is needed and the PQR field is only 1 bit long. If the R bit is '1', then the 16x16 block is subdivided into four 8x8 blocks as shown in Figure 9B, and at least four additional bits will exist in the PQR field.

[0046] The additional four bits are referred to as 'Q' information. Each bit of Q denotes a subdivision of an 8x8 block into four 4x4 blocks. For each bit of Q that is set, four

more bits of 'P' are present to indicate if any of the 4x4 blocks are subdivided into 2x2. Accordingly, the length of PQR data can be 1 to 21 bits long, depending on the block size assignment within the 16x16 block. If every 8x8 block is subdivided, then the PQR information will be 21 bits in length. Figure 10A-D shows some examples of the 16x16 blocks with corresponding PQR data.

[0047] Accordingly, each block may be divided into sub-blocks of sizes 8x8, 4x4, and or 2x2 depending on the assignment criterion. The criterion to subdivide an nxn block is the block variance as follows.

$$Block\_variance = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}x^2(m,n) - \left[\frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}x(m,n)\right]^2$$

[0048] An nxn block of pixels shall be subdivided into 4 n/2xn/2 sub-blocks if the block variance of the nxn block exceeds a certain threshold. Here, the block mean can have values in the range (0, 1023) for a 10-bit image. Thus, the image is divided into 12 bins and a set of thresholds is used for each bin for each color component. Also, the thresholds may be determined based on the statistics collected from a number of image frames of different types. A typical threshold set is shown in Figure 12A-C.

[0049] Figure 11 illustrates an example process 1100 for generating the PQR information for a 16x16 block is shown. For each block, the mean value and variance V16 is obtained (1110). The variance V16 is compared with the appropriate threshold T16 for the corresponding mean value (1115). If the variance V16 is not greater than threshold T16, the R value of the PQR data is set to 0 and the process ends (1120). Otherwise, the R value is set to 1 (1125). The variance V8(i), {i = 1 to 4} is then obtained for each of the four 8x8 subblocks 0 to 3 as shown in Figure 9B and each variance V8(i) is compared with the appropriate threshold T16 to determine the Q values for the PQR data (1130 to1140). If a variance V8(i) is not greater than threshold T8, the corresponding Q(i) is set to 0 (1145). Otherwise, the Q(i) value is set to 1 (1150). The variance

V_4(j), {j = 1 to 4} is then obtained for each of the four 4x4 subblocks of each 8x8 block for which Q(i) is set to 1 and each variance V_4(j) is compared with the appropriate threshold T_4 to determine the P values for the PQR data (1155 to1165). If a variance V_4(j) is not greater than threshold T_4, the corresponding Q(j) is set to 0 (1170). Otherwise, the Q(j) value is set to 1 (1175).

[0050] Thus, the PQR information may be generated and used for deblocking in image decompressor such as image decompressor 200. The PQR information is used to determine edge content in the image. The greater the edge information in a block, the smaller the block size and the longer the PQR code. Figure 14 shows an example process 1400 for processing images compressed using ABSDCT and Figure 15 shows the variable definition used in process 1400.

[0051] When determining whether deblocking filter is to be used for two neighboring blocks, the PQR information is obtained for each block (1410). If both PQR bits are greater than 5 bits (1415), the process ends. Namely, both blocks are determined to be subdivided and deemed to contain sufficient edge information. Otherwise, if one of the PQR bits is greater than 5 bits, a two point averaging filter is used on {x1, y1} (1420 and 1425). If neither of the PQR bits is greater than 5 bits, then difference values d1, d2 and d3 are obtained (1430). If d1, d2 and d3 are greater than threshold TD, then a 6 point Gaussian filter is used on {x1, x2, x3, y1, y2, y3} (1435 and 1440). If d1 and d2 are greater than threshold TD, then a 4 point Gaussian filter is used on {x1, x2, y1, y2} (1445 and 1450). If d1 is greater than threshold TD, then a two point averaging filter is used on {x1, y1} (1455 and 1460).

[0052] In process 1400, the embodiment is not limited to an averaging filter and/or Gaussian filter. Various filters may be used other than the averaging filter and/or Gaussian.

[0053] As shown, a deblocking filter module can easily be implemented in a decompressor. Accordingly, artifacts may be significantly mitigated and the visual quality of an image is improved. Note that while deblocking filter module 240 is shown to be implemented separately from inverse VBSDCT module 230 and from processor 250, one or a combination of deblocking filter module 240, inverse VBSDCT module 230 and processor 250 may be implemented together.

[0054] Also, the embodiments may be implemented by hardware, software, firmware, middleware, microcode, or any combination thereof. When implemented in software, firmware, middleware or microcode, the elements of the embodiment are the program code or code segments to perform the necessary tasks may be stored in a machine readable medium (not shown). A code segment may represent a procedure, a function, a subprogram, a program, a routine, a subroutine, a module, a software package, a class, or any combination of instructions, data structures, or program statements. A code segment may be coupled to another code segment or a hardware circuit by passing

and/or receiving information, data, arguments, parameters, or memory contents. Information, arguments, parameters, data, etc. may be passed, forwarded, or transmitted via any suitable means including memory sharing, message passing, token passing, network transmission, etc. Also, the machine readable medium may be implemented in an article of manufacture for use in a computer system and may have machine readable code means embodied in therein.

[0055] In addition, although the embodiments have been described using variable block size DCT, the deblocking technique as described above can also be implemented in DCT with fixed block sizes. In such cases, the BSA information would be generated, but would not used for actual DCT. Instead, the BSA information would be used at the image decompressor to determine whether deblocking is necessary for two neighboring blocks.

[0056] It should be noted that the foregoing embodiments are merely examples and are not to be construed as limiting the invention. The description of the embodiments is intended to be illustrative, and not to limit the scope of the claims. As such, the present teachings can be readily applied to other types of apparatuses and many alternatives, modifications, and variations will be apparent to those skilled in the art.

What is claimed is: